

JP4308971 A
BINARY SEARCH MEMORY
NIPPON STEEL CORP

Abstract:

PURPOSE: To speedily retrieve stored data with configuration like a hardware without using a CPU by providing an exclusive circuit to execute the binary search of the data in respect to a memory storing the data in the order of numerical values based on an identification value to identify the data.

CONSTITUTION: A memory 10 stores the data composed of the part of a data main body and the part of a key corresponding to this part of the data main body by one-to-one in an ascending order along the continuous absolute addresses of the memory 10. In this case, when two keys are compared, the data are stored so that the key of a smaller value can be stored at a smaller address and the key of a larger value can be stored at a larger address. A sense amplifier 12 reads out keys and data from the memory cell 10, and a decoder 14 outputs the keys and the data through a data bus 16 to a control circuit 32 and selects the specified memory in the memory cell 10. Thus, the data in the memory can be easily retrieved.

COPYRIGHT: (C)1992,JPO&Japio

Inventor(s):

OKAWA YOSHINORI

Application No. JP1991101868A **Filed** 19910406 **Published** 19921030

Original IPC(1-7): G06F001540

Priority:

JP1991101868A 19910406

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平4-308971

(43) 公開日 平成4年(1992)10月30日

(51) Int.Cl.⁵
G 0 6 F 15/40識別記号 庁内整理番号
S 0 0 B 7056-5L

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 5 頁)

(21) 出願番号 特願平3-101868

(22) 出願日 平成3年(1991)4月6日

(71) 出願人 000006655

新日本製鐵株式会社

東京都千代田区大手町2丁目6番3号

(72) 発明者 大川 吉寛

東京都千代田区大手町2丁目6番3号 新

日本製鐵株式会社内

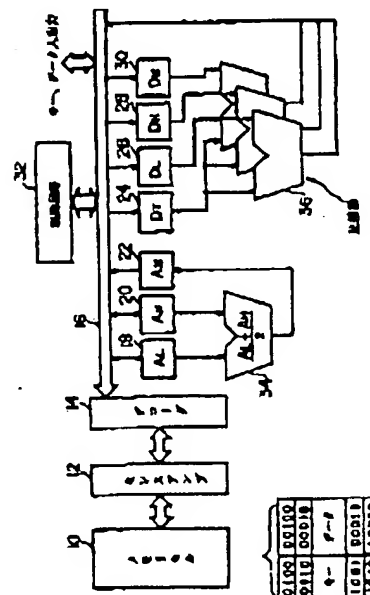
(74) 代理人 弁理士 半田 昌男

(54) 【発明の名称】 バイナリサーチメモリ

(57) 【要約】

【目的】 メモリに格納されたデータの検索をCPUを用いずにハードウェア的な構成で迅速に行うこと。

【構成】 メモリのデータに対して、識別値に基づいて、この識別値の数値順に記憶しておき、データのバイナリサーチを行う。



(2)

特開平4-308971

(2)

特開平4-308971

【特許請求の範囲】

【請求項1】 複数のデータを、これらのデータを識別する識別値に基づいて、数値順に記憶したメモリに対して、前記データのバイナリサーチを行う専用の回路を設けたことを特徴とするバイナリサーチメモリ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、記憶された多数のデータの中から目的とするデータを高速に検索することができるバイナリサーチメモリに関するものである。

【0002】

【従来の技術】 メモリに記憶されたデータの中から希望するデータを検索する場合、従来の方法では、まず図4に示すようにCPU（中央処理装置）50からメモリ52をアクセスしてアドレスの順序に従って各データに付加された識別信号であるキーを読み出す。次に、予めCPU50内に入力してある検索したいデータのキーとメモリ52から読み出したキーとを逐一照合し、一致するものを捜す。そして、一致するキーが見つかったときはそのキーに対応するデータを希望するデータとして読み出し、以後に必要な処理に使用する。

【0003】

【発明が解決しようとする課題】 しかしながら、従来のようにデータが見つかるまで次々とデータを読み出す方法で検索を行うと、検索したいデータがメモリアドレスの最後の方に格納されている場合には、データの検索に要する時間が非常に長くなる。しかも、読み出したキーと検索すべきキーとが一致するかどうかをCPU50内部の演算回路を用いてソフトウェア的に行っているの

で、ハードウェア的に処理する場合に比べて処理速度が遅い。

【0004】 また、CPU50には図4に示す以外にも種々の処理を行わせるのが一般的であり、データの検索はCPU50が行う処理のうちのほんの一部に過ぎない。しかし、データの検索を行っている期間中はCPU50をそれ以外の別のタスクの処理に使用することができないため、処理能率が悪く、結果として全体の処理が遅延するという問題がある。

【0005】 本発明は上記事情に基づいてなされたものであり、全体の処理を制御するCPUを用いずにデータの検索ができ、しかもハードウェア的な構成で迅速なデータ検索が可能となるバイナリサーチメモリを提供することを目的とするものである。

【0006】

【課題を解決するための手段】 上記の目的を達成するための本発明に係るバイナリサーチメモリは、複数のデータを、これらのデータを識別する識別値に基づいて、数値順に記憶したメモリに対して前記データのバイナリサーチを行う専用の回路を設けたことを特徴とするものである。

【0007】

【作用】 本発明は前記の構成によって、メモリ内のデータの検索を行うので、データ検索を行っている期間中でもあってもCPUを別の処理のために使用することができる。また、データの検索を専用構成した回路によってハードウェア的に行うので、データ検索に要する時間が短縮される。

【0008】

【実施例】 以下に図面を参照しつつ本発明の一実施例について説明する。図1は本発明の一実施例であるバイナリサーチメモリのブロック図、図2はメモリの内容を簡単化して図1に示すバイナリサーチメモリの動作を分かりやすく説明するための図、図3は音響機器の音調制御に本発明のバイナリサーチメモリを適用した例を示すブロック図である。

【0009】 図1において、メモリ10にはデータ本体の部分及びこのデータ本体の部分と1対1に対応するキーの部分よりなるデータ（以下単にデータといったときはこのデータのうちのデータ本体の部分を目指すものとする）が、図示する様にこのキーに基づいてメモリ10の連続する絶対番地に沿って昇順に記憶されている。すなわち、2つのキーの値を比較したときに、キーの値が小さい方が小さいアドレスに、キーの値が大きい方が大きいアドレスになるよう記憶されている。

【0010】 センスアンプ12は、メモリセル10からキー及びデータの読み出しを行いデコード14は、データバス16を介して後述する制御回路32及びデータを出力するとともに、メモリセル10の中の特定のメモリの選択を行う。

【0011】 制御回路32はキー及びデータの流れを制御する回路であり、データバス16を介してキー及びデータを各レジスタ18、20、22、24、26、28、30に入力するものである。レジスタ24は、検索しようとするデータのキーの値を記憶するレジスタである。レジスタ26はメモリセル10のキーの中で最小のキーの値を記憶するレジスタであり、またレジスタ18は、このレジスタ26のキーに対応するメモリアドレスの値が記憶されている。

【0012】 レジスタ28は、メモリセル10のキーの中で最大のキーの値を記憶するレジスタであり、レジスタ20は、このレジスタ28のキーに対応するメモリアドレス値を記憶するレジスタである。

【0013】 演算回路34は、レジスタ18、レジスタ20のメモリアドレスの値の平均値を算出し、レジスタ22に出力する。レジスタ30は、レジスタ22のメモリアドレスの値を記憶するものである。そして、比較器36は、レジスタ24に記憶されている、キーの値を各レジスタ26、28、30のキーの値とを比較し、その結果をデータバス16を介して制御回路32に出力す

50 る。

(3)

特開平4-308971

(3)

特開平4-308971

3

【0014】次に、図1の回路の動作について説明する。ここで、検索しようとするデータのキーを D_i 、キーの中で最小の値を D_i 、このキー D_i のデータが記憶されているメモリアドレスを A_i 、キーの中で最大の値を D_j 、このキー D_j のデータが記憶されているメモリアドレスを A_j とする。また $(A_i + A_j) / 2$ で表されるアドレス（小数が現れる場合には小数点以下は切り捨てるか、または切り上げる）を A_k 、このアドレス A_k に記憶されているデータのキーを D_k とする。

【0015】図1において、まずメモリ10から読み出した最小のキー D_i をレジスタ26に、最大のキー D_j をレジスタ28に、アドレス A_i をレジスタ18に、アドレス A_j をレジスタ30にそれぞれ入力する。次に、演算回路34において $(A_i + A_j) / 2$ を計算し、得られた値 A_k をレジスタ22に入力するとともにアドレス A_k に記憶されたデータのキー D_k をメモリ10より読み出してレジスタ30に入力する。そして、比較器36において D_i と D_k 、及び D_k と D_j をそれぞれ比較する。このとき、仮に $D_i = D_k$ （上記で $(A_i + A_j) / 2$ の値を求めるときに小数点以下を切り捨てる場合に対応する）又は $D_k = D_j$ （上記で $(A_i + A_j) / 2$ の値を求めるときに小数点以下を切り上げる場合に対応する）となっている場合には、検索しているキー D_i はメモリ10の中にはないと判断され、入出力回路より特殊なデータとして全ビット「0」又は全ビット「1」のデータが出力される。

【0016】 $D_i = D_k$ でも $D_k = D_j$ でもない場合には、 D_i と D_k とを比較する。ここで $D_i = D_k$ であればこの D_i が求めているキーであり、このキーに対応するデータをメモリ10のアドレス A_i から読み出して出力する。次に $D_i > D_k$ の場合には、 D_i を新たに D_i としてレジスタ28へ入力するとともに、 A_i を新たに A_k としてレジスタ20に入力する。そして、上記と同様の手順を繰り返す。一方、 $D_i < D_k$ の場合には、 D_k を新たに D_i としてレジスタ26へ入力するとともに、 A_k を新たに A_i としてレジスタ18に入力し、同じく上記と同様の処理を繰り返す。こうして最終的に $D_i = D_j$ となる D_i が見つければ、これが求めているデータのキーであるので、対応するメモリセルよりそのデータを読み出す。一方、最終的に $D_i = D_k$ 又は $D_k = D_j$ となった場合には求めている値がメモリ10の中にはないことが明らかとなり、検索作業は終了する。

【0017】次に、上記の動作をより分かりやすくするために図2に示す簡略化した具体例について説明する。図2において2列に構成した表は、左側が昇順に配列されたデータのキー、右側が対応するデータが記憶されているアドレスとする。この中で例えば4という値のキーを検索する場合を考える。ここではまず D_i は1、 D_j は1000であるので、これらを図1のメモリ10から読み出してレジスタ26、28にそれぞれ格納する。ま

4

た、 A_i は10、 A_j は22であるので、 A_k は16となりこのアドレスにおけるキーが $D_k = 50$ であることが分かる。ここで、 $D_i = 50$ と $D_k = 4$ を比較すると $D_i > D_k$ なので、今度は D_i を50、 A_i を16とする。そして上記と同様に再び A_k の値を求めると $A_k = 13$ となり、このアドレスにおけるキーの値9が D_k となる。ここでまた $D_i = 9$ と $D_k = 4$ を比較すると $D_i > D_k$ であるので、 D_i を9、 A_i を13とする。このとき A_k を求めると11.5となるが、小数点以下の切り捨てを行って $A_k = 11$ とする。このアドレスにおけるキーの値 $D_k = 4$ と、求めるキーの値 D_i とを比較すると等しいので、メモリのこの D_k に対応するアドレスから読み出しを行うことによって求めるデータが得られる。

【0018】次に、図2において $D_i = 10$ という値のキーを検索する場合を考える。最初は上の例と同様に $A_i = 10$ 、 $A_j = 22$ であり、 $A_k = 16$ である。ここで $D_i = 50$ と $D_k = 10$ とを比較すると $D_i > D_k$ であるので、 A_i を新たに16として同様の操作を行う（ A_i は10のまま）。このとき同じようにして A_k をもとめると $A_k = 13$ となり D_k を求めると9になるが、この値は $D_i = 10$ よりも小さいので、この場合には13を新たに A_i として代入する（ A_i は16のまま）。そして再び A_k を求めると14.5となるが、小数点以下を切り捨てて $A_k = 14$ とする。このときの $D_k = 12$ は $D_i = 10$ よりも大きいので、今度は A_i として14を代入する（ A_i は13のまま）。この A_i と A_k から A_k を求めると13.5となるが小数点以下を切り捨てて13とする。このときの $D_k = 9$ は、同時に $D_i = 9$ でもあるので、図2の表の中には $D_i = 10$ という値のキーは存在しないと判断され、図2を見ると分かるように事実存在していない。図2では分かりやすくするためにデータの数が極端に少ない場合について説明したが、実際には図1の回路は通常膨大な量のデータを検索する場合に適用される。

【0019】図3は物理的性質が時間的に変化する音場空間の音場制御に図1に示すバイナリサーチメモリを用いた場合のブロック図である。例えば自動車の中で音楽を聞いているような場合には、周囲からは多くの雑音が入ってくるので、音場空間の物理的性質のひとつである車内のトータルの音量は時々刻々変化している。そこで、この雑音をマイクなどで取り込んで、スピーカから出ている音楽の音量をその雑音レベルに応じて適宜に変化させることができれば、車内にいる人間に聞こえる音楽に関する音量については等価的に一定に保つことができる。

【0020】そこで、図3に示すように、音源49から音楽などの信号Sと周囲からの雑音Nとが混ざった信号をマイク40によって取り込み、この信号S+Nと前段増幅器46からの信号Sとを減算回路42において減算

(4)

特開平4-308971

(4)

特開平4-308971

5

する。こうして分離された雑音成分Nを図1で説明したキーとしてパラメータ抽出回路44に与える。パラメータ抽出回路44には図1のバイナリサーチメモリが備えてあり、そのメモリ内部には予め後段増幅回路48の増幅率を格納しておく。そして、この格納されている増幅率と1対1に対応させてあるキーを雑音レベルに比例させておき、ある雑音Nが入ってきたときには、そのキーに対応する所定の増幅率のデータを後段増幅回路48に出力する。これにより、後段増幅回路48は、所定の増幅率で音源49からの信号Sを増幅してスピーカ50から所定の音量で信号Sが報音される。よって、車内において聴取者が等面的に一定の音量で音楽などの音声信号を聞くことができるよう音場空間を制御することができる。また、可聴周波数をいくつかの範囲に分け、それぞれの周波数範囲毎に別々に音量の制御を行うようにすれば、より精細な音場制御を行うことができるが、本発明のようなバイナリサーチメモリを用いれば、高速な処理が可能となるので、低価格にも対応することができる。

【0021】上記のような音場空間の制御を行う場合に、従来のようにソフトウェア的に全てのデータを検索していたのでは、処理速度が遅く十分な処理はできない。しかし、図1に示す本実施例のバイナリサーチメモリを用いることによって、データの検索処理は専用の制御回路によってハードウェア的に行われるので処理が高速になり、適切な音場空間の制御が可能となる。

【0022】

【発明の効果】以上説明したように本発明によれば、メモリにバイナリサーチを行う専用の回路を設けたことにより、種々のタスクを処理するCPUをデータの検索処理のために使用しないですむので、処理能率の向上を図ることができ、しかもデータの検索をハードウェア的に

6

高速に処理することが可能になり、したがって例えば、音場空間の物理的性質が時間的に変化する場合の音場制御のように、膨大な量のデータの中から必要とされるデータを高速に検索しなければならない場合などに最適なバイナリサーチメモリを提供することができる。

【図面の簡単な説明】

【図1】本発明の一実施例であるバイナリサーチメモリのブロック図である。

【図2】メモリの内容を簡略化して図1に示すバイナリサーチメモリの動作を分かりやすく説明するための図である。

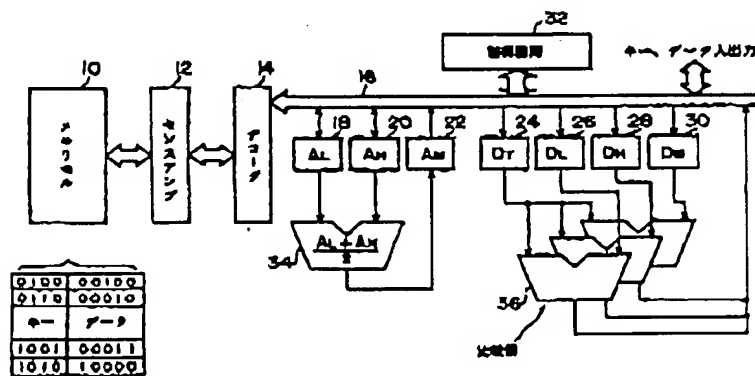
【図3】音響機器の音場制御に本発明のバイナリサーチメモリを適用した例を示すブロック図である。

【図4】ソフトウェア的にデータを検索する従来の回路を概略的に示したブロック図である。

【符号の説明】

- 10、52 メモリ
- 12 センスアンプ
- 14 デコーダ
- 16 データバス
- 18、20、22、24、26、28、30 レジスタ
- 32 制御回路
- 34 演算回路
- 36 比較器
- 40 マイク
- 42 演算回路
- 44 パラメータ抽出回路
- 46 前段増幅回路
- 48 後段増幅回路
- 50 CPU

【図1】



【図2】

1	10
4	11
8	12
9	13
12	14
20	15
50	16
...	...
1000	22

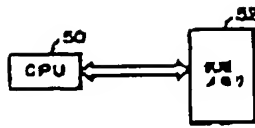
(5)

特開平4-308971

(5)

特開平4-308971

【図4】



【図3】

